

CLAIMS

What is claimed is:

1. A data service system, comprising:

a plurality of duplicate content servers for hosting a content file with a file name, each content server storing a version of the same content file that is referred to by a file reference;

an updating engine that, when receiving an update of the content file, generates an updated version of the content file;

a file name binding server coupled to the updating engine and the content servers to generate a new file reference for the updated version of the content file, wherein the file name binding server updates each of the content servers with the updated version by sending the new file reference to each of the content servers.

2. The data service system of claim 1, further comprising a content store that stores the updated version of the content file before the updated version is fetched into the content servers.

3. The data service system of claim 1, wherein each of the content servers further comprises

a cache that stores a version of the content file;

a file reference binding table that stores the new file reference of the updated version of the content file, wherein the new file reference is also associated with the file name of the content file;

a file manager that, when receiving a request for the content file with

the file name, accesses the binding table to retrieve the new file reference using the file name, and then accesses the cache with the new file reference.

4. The data service system of claim 3, wherein the file manager verifies that the content retrieved is indeed the updated version of the content file by applying a hash algorithm to the retrieved content and then determining if its outcome matches the new file reference.

5. The data service system of claim 4, wherein if the outcome of the hash algorithm does not match the new file reference, the file manager accesses the cache again with the new file reference for the updated version of the content file.

6. The data service system of claim 3, wherein if a cache miss is resulted from the access to the cache with the new file reference, the file manager then fetches the updated version of the content file from an external content store that is coupled to the updating engine.

7. The data service system of claim 3, wherein each of the content servers further comprises an HTTP engine that extracts the file name of the content file from the request and sends the file name to the file manager.

8. The data service system of claim 3, wherein the content stored in the binding table in each of the content server is the same as that stored in the file name binding table server.

9. The data service system of claim 1, wherein the updating engine causes the file name binding table server to generate the new file reference when the updating engine generates the updated version of the content file.

10. In a data service system having a plurality of content servers hosting a content file with a file name, each content server comprising:
a cache that stores a version of the content file;
a file reference binding table that stores a file reference of a version of the content file, wherein the file reference is associated with the file name of the content file;

a file manager that, when receiving a request for the content file with the file name, accesses the binding table to retrieve the file reference using the file name, and then accesses the cache with the file reference for that version of the content file corresponding to the file reference.

11. The content server of claim 10, further comprising an HTTP engine that extracts the file name of the content file from the request and sends the file name to the file manager.

12. The content server of claim 10, wherein the file manager verifies that the content retrieved is indeed the particular version of the content file referenced by the file reference by applying a hash algorithm to the retrieved content and then determining if its outcome matches the new file reference.

13. The content server of claim 12, wherein if the outcome of the hash algorithm does not match the file reference, the file manager accesses

the cache again with the file reference for that version of the content file.

14. The content server of claim 10, wherein if a cache miss is resulted from the access to the cache with the file reference, the file manager then fetches the version of the content file from an external content store.

15. The content server of claim 10, wherein the file reference binding table is updated with a new file reference when an update of the content file is received by the data service system, wherein the new file reference corresponds to the updated version of the content file.

16. The content server of claim 15, wherein the updated version of the content file is stored in a content file external to the content server.